In Proceedings of the Eleventh International Conference on Social Robotics, pp. 133-143 2019. Springer.

Optimal Use Of Verbal Instructions For Multi-Robot Human Navigation Guidance

Harel Yedidsion, Jacqueline Deans, Connor Sheehan, Mahathi Chillara, Justin Hart, Peter Stone, and Raymond J. Mooney

Department of Computer Science, The University of Texas at Austin
harel@cs.utexas.edu

Abstract. Efficiently guiding humans in indoor environments is a challenging open problem. Due to recent advances in mobile robotics and natural language processing, it has recently become possible to consider doing so with the help of mobile, verbally communicating robots. In the past, stationary verbal robots have been used for this purpose at Microsoft Research, and mobile non-verbal robots have been used at UT Austin in their multi-robot human guidance system. This paper extends that mobile multi-robot human guidance research by adding the element of natural language instructions, which are dynamically generated based on the robots' path planner, and by implementing and testing the system on real robots.
Generating natural language instructions from the robots' plan opens up a variety of optimization opportunities such as deciding where to place the robots, where to lead humans, and where to verbally instruct them. We present experimental results of the full multi-robot human guidance system and show that it is more effective than two baseline systems: one which only provides humans with verbal instructions, and another which only uses a single robot to lead users to their destinations.

Keywords: Multi Robot Coordination · Natural Language · Human Robot Interaction · Indoor Navigation

1 Introduction

Finding one's way in an unfamiliar office building, university, or hospital can be a daunting task. Even when provided with navigational instructions, the lack of an indoor localization system, and the lack of reliable pedestrian odometry makes it difficult for humans to successfully follow them. Following instructions is efficient for short sequences, but memorizing a long sequence of instructions is difficult, and the tendency to make mistakes increases with the length of the instruction sequence and the complexity of the environment.

As service robots become increasingly abundant in large buildings [12, 4], and steadily more capable of autonomous navigation, there has been growing interest in using them to guide humans in buildings. However, for safety reasons, even state-of-the-art service robots still travel much slower than the average human; therefore, following them is reliable but tedious. Additionally, robots have difficulties changing floors and opening doors.

To mitigate these problems, we developed a system of mobile robots that provides guidance to newcomers to the GDC building at UT Austin. Once a visitor approaches one of the robots and requests to reach a goal location, the system calculates the shortest path to the desired destination, and then, based on the characteristics of each region of the path, decides where a robot will lead the person and where verbal instructions will be provided. The characteristics that are considered for each region are: (i) the length of the path through that region, (ii) the region's traversability by a robot, and (iii) the probability of a human going wrong there.

The use of multiple robots allows for separate intervals of leading and instructing within a single navigational path. The robots that are used in this research are the BWIBots (See Figure 1), a custom-built fleet of robots that are part of the Building Wide Intelligence (BWI) project [8]. The generation of natural language navigational

Fig. 1. Three BWIBots used in this study.

instructions is based on the robot's planned path and a map annotated with landmarks.

We empirically tested the system on human participants who were not familiar with the GDC building, and measured the time it took them to reach a given goal location. Empirical results show that compared to baselines of instructions only and leading only, the combination of leading and instructing provides the best results in terms of minimizing the time to destination and maximizing the success rate.

The remainder of the paper is structured as follows. Section 2 reviews related work. Section 3 formally defines the problem, and Section 4 outlines our approach to optimally solving it. Section 5 describes the technical details of implementing of the system on the BWIBots. Section 6 presents the natural language instruction generation module. Section 7 details the experimental evaluation and discusses the results, and Section 8 concludes.

2 Related Work

Previous studies of the multi-robot human guidance problem developed a central system whose goal is to guide a person to the destination efficiently, while limiting the robots' time away from their background tasks [6], and later extended it to multiple concurrent guidance tasks [7]. Both of these systems did not make use of natural language instructions, but rather used arrows on the screen to indicate the position of the next robot the human should go to. We extend this research by adding natural language instructions, and by applying and testing the system on real robots.

Another relevant project used a single stationary robot to give verbal instructions inside a building [1]. The paper provides some valuable insights and directions for future work regarding the effectiveness of the robot's generated instructions. The authors discuss the challenge of communicating long paths, which result in directions that are difficult for the listener to understand and retain. We propose to mitigate this challenge by using multiple mobile robots which enable breaking up long instruction sequences to shorter, and more user-friendly ones. This paper ([1]), and others [5], highlight the use of landmarks as navigational waypoints – a technique we leverage in our work.

Generating natural language navigational instructions has been attempted using a Seq-to-Seq network that was trained on a dataset of human generated instructions for navigating a grid-world domain [2]. Other approaches such as the top performing one in the GIVE 2.5 challenge [10], a competition for natural language generation systems that guide human users through solving a task in a virtual environment, use a template-based method [3]. We use a similar template-based method for generating natural language instructions.

3 Problem Definition

Our problem formulation is similar to that of the Multi-Robot Human Guidance (MRHG) problem [6]. A MRHG problem begins when a human approaches a robot and requests assistance to reach a destination in the building, and ends when the destination is reached. At our disposal we have a team of mobile robots that can autonomously navigate the building. The problem we are studying is how to efficiently utilize them to guide humans to their desired destination in the building as quickly as possible.

Unlike Khandelwal et al. [6], where the set of available guidance actions were either *Direct* (Display a directional arrow on the display interface of a robot) or *Lead* (Have the human follow a robot as it navigates to a required location), in our system the robots are augmented with the ability to generate and vocalize natural language instructions and therefore use the action *Instruct* instead of *Direct*.

Our primary objective is to minimize the time it takes the human to reach the destination. Khandelwal et al. [6] also considered minimizing the robots' time away from their background tasks as a secondary objective, which we do not actively try to optimize for. However by simply supplementing the robots with the *Instruct* action, the system is able to reduce the robots' leading time and consequently improve this secondary objective as well.

The notations we use and assumptions we make are as follows. The environment, E, in which guidance assistance is required is a fully connected space which is divided into a set of non overlapping domains D such that $E = \bigcup_{i=1}^{|D|} d_i$. Each domain, $d_i \in D$ has one robot $r_i \in R$ assigned to perform a background task in that domain.

A guidance task, $T(o, g)$ is a request by a human to get from origin location o to goal location g. We assume that at the time a guidance task is created, the human and one robot are co-located at origin o.

The robots can communicate with each other and announce that a new guidance task needs to be performed. We assume that the environment is divided into domains which are small enough to enable the robots to get into the required position in their domain in time to support a high priority guidance task, once it is communicated. The robots can navigate the environment autonomously with an average speed of v_r. We assume that humans travel at an average speed of v_h, and that $v_h > v_r$.

We assume that robots have a map of the environment and can use it to plan the shortest path, p, from origin to goal. The map is divided into a set of regions, L, which correspond to rooms, corridors, elevators, and open spaces, each with attributes of: physical dimensions, neighboring regions, how traversable this region is to a robot, and what is the probability of a human going wrong there while following navigational instructions. Each domain contains several regions. We denote the subset of regions of L that contain path p as L^p.

The robots can reach any given location reliably, i.e., do not make any wrong turns. However, for the robots, some regions are more traversable than others, e.g, the elevator, or the very busy regions are difficult to navigate through. We denote the traversability of region l_j for a robot as trv_j. The harder a region is to traverse, the lower its trv value will be.

Regarding humans, we assume and validate the following assumption empirically: given a set of navigational instructions, humans have some probability of going wrong in every region, and that probability increases as the length of the instruction sequence increases. We denote the inherent navigational complexity of region l_j for a human as cmp_j. The dynamic parameter that represents the number of previously consecutive instructed regions before l_j is denoted as cir_j. The actions that a robot can take to guide a human are:

- *Lead* - The robot asks the human to follow it as it navigates to a desired destination.
- *Instruct* - The robot generates and vocalizes natural language instruction from its current location to a desired destination.

The robots chose one action per region.

We assume that action durations are not fixed and are dependent on the state i.e, the complexity of the region and the number of consecutive previously instructed regions. We also assume that when navigating to a location in a building, humans will eventually reach their destination even if they take a wrong turn, at the expense of longer travel time. This assumption allows us to trade off state transition probabilities with action durations, i.e., if a human is given navigational instructions from origin to goal, and the regions in the path have some probability of going wrong, the human will reach the goal successfully but the time it takes to traverse these regions will be directly affected by the respective probabilities of going wrong. We assume that the time required to say the instructions for a single region is constant and equals t_c.

Thus, the MRHG can be formulated as a stochastic planning algorithm where the goal is to find the plan that results in the shortest time in expectation.

Formally, the MRHG problem is: Given a guidance task $T(o, g)$ in environment E, populated by a set R of robots, each located in its respective domain, find an optimal sequence of actions that will minimize the human's expected time to reach the destination goal g. Denoting t_j as the time estimation per region l_j in the shortest path p from o to g:

$$min \sum_{l_j \in L^p} t_j \quad S.T.$$

- Only one consecutive *lead* sequence per robot.
- The robot can use the *lead* action only at the start of its guidance sequence.
- A robot can only lead in its domain, but can instruct through other domains.
- When a robot instructs to a region in another robot's domain, a transition occurs and requires the robot in that domain to navigate to that region and wait for the human there to start its portion of the guidance task.
- The number of transitions is limited to the number of domains (and robots) in the path.

4 Optimal MRHG Problem Solver

In this section we outline our approach to optimally solve a guidance task $T(o, g)$ for the MRHG problem. First, by using the robot's planner we calculate the shortest path p from o to g. Second, we calculate for each region in the path, $l_j \in L^p$, the length of the path through that region $length(p, l_j)$. Third, the solver calculates a time estimation for each possible combination of *Lead/Instruct* for each robot. In order to calculate the time estimate for a path, we dynamically calculate the additional time per region that is a result of it being in a long chain of consecutive instructed regions. The parameter that represents the number of previously consecutive instructed regions for region l_j is denoted as cir_j. Since a robot is located in transition regions, the probability of going wrong there is reduced by a factor denoted as the robot observably factor, rof. The time estimation per region, t_j, is calculated as follows:

$$t_j = \begin{cases} (length(p, l_j)/v_r)/trv_j & \text{if } action = Lead \\ (length(p, l_j)/v_h) \cdot cmp_j \cdot (cir_j + 1) + t_c & \text{if } action = Instruct \\ (length(p, l_j)/v_h) \cdot cmp_j \cdot (cir_j + 1)/rof + t_c & \text{if } Transition \quad \text{at} \quad l_j \end{cases}$$

For a region where a *Transition* occurs, the probability of going wrong is reduced since we place a robot at the intersection of the two regions and it is harder to miss it as opposed to other less noticeable landmarks.

Finally, the solver chooses the action combination that results in the shortest time. It runs a branch-and-bound search on the tree of possible combinations saving the current minimal combination time and abandoning combinations that result in longer times. This approach speeds up computation compared to brute force search and allows for real-time optimization of paths with up to 10 regions, i.e., without creating an awkward pause in the dialog.

5 Robot Implementation

In this section, we briefly describe the hardware design of the BWIBots [8] used in this study. The robotic platform is built on top of the Segway RMP mobile base. The robots are equipped with a Hokuyo lidar (for navigation and obstacle avoidance), a Blue Snowball microphone, and a speaker (for conducting the dialog). One of the robots is equipped with a Kinova MICO arm (useful for gesturing the initial orientation of the very first sequence).

On the non-armed robots, a Dell Inspiration computer executes all necessary computation and doubles as a user interface. For the arm robot, an Alienware computer is used. The robots' mobile Segway base is reinforced with additional 12V Li-Ion batteries to power the base, arm, computer, and sensors for up to 6 hours of continuous operation. The software architecture used on the BWIBots is built on top of the Robot Operating System (ROS) framework [9]. It includes a custom built layered architecture that spans high level planning to low level motion control and allows autonomous navigation in our department building.

Multi-robot communication is achieved using the *rosbridge_suite* and *Node.js* packages running on a central server, and passing ROS messages between the different ROS masters that manage each individual robot. Robot-spoken instructions are delivered using Google's WaveNet text-to-speech [11]. [1]

6 Generating Natural Language Instructions

In this section we present the system for generating natural language instructions for indoor navigation. Given a starting point, ending point, and an annotated map, our system produces landmark-based instructions to a goal location. These instructions are generated by translating the robot's planned path into an intermediate abstract syntax, and then into natural language, using a template-based method [3]. In order to validate the quality of the generated instructions, we conducted a preliminary human study comparing our system's generated instructions to human-generated ones. For human directions, we asked a student who was very familiar with the building, but not with our system, to generate instructions as if someone had asked him how to get there.

Twenty five participants each traveled to four different locations on the GDC third floor after receiving random combinations of human/robot-generated instructions. After each trial, participants answered six questions, each on a 6-point semantic differential scale, each measuring one metric of the quality of the instructions. The metrics were: understandability ("How easy was it to understand the instructions?"), memorability ("How easy was it to remember the instructions?"), informativeness ("How much information did the instructions provide?"), efficacy ("How easily did you arrive at the destination?"), usefulness of landmarks ("How helpful were the landmarks provided in the instructions?"), and naturalness ("How natural did the instructions sound?").

[1] The BWI code is open source and can be found here: https://github.com/utexas-bwi.

Our system was ranked within 0.75 points of the human generated instructions for each metric, with no statistically significant differences found for understandability (p=0.089) memorability (p=0.367) and informativeness (p=0.289). Most importantly, we timed all participants through each trial and found no statistically significant differences between human- versus robot-generated instructions, according to two-sample t-tests. Thus, following the results of this preliminary system validation, we concluded that our instruction generating module provides sufficiently effective instructions to be used in the full MRHG system.

7 Experiments

Setup - To study the performance of this system, we chose a typical navigation task facing a newcomer to the building. The path starts from the GDC entrance hall and ends in front of an office on the third floor. We recruited 30 participants with no prior knowledge of the building, (12 male, 18 female, ages 12-58), from the UT Austin campus by fliers, email, and word of mouth. We conducted a two-condition inter-participant study contrasting the MRHG condition to the *Instructions* condition, defined as follows. The participants were divided into two groups. The first 15 were only given instructions by the robot and had to follow those instructions to the best of their abilities. The results from this group were used both as a benchmark for the full multi-robot system, and as a means of tuning the parameters for the probability of people going, based on the frequency of that occurrence in each region. These parameters were later used by the MRHG optimization module to determine where to lead and where to instruct. The other 15 participants interacted with the full *MRHG* system which utilized 3 robots to lead and instruct the humans to the goal. As another benchmark, we timed a single robot as it navigated from the origin to the destination, averaged over 15 runs. This was done to simulate a *Leading* only guidance solution. A visualization of the path can be seen in Figure 2 and a demo video of the BWIBots performing a MRHG task with verbal instructions can be viewed here https://youtu.be/5MSdMwfw6QI.

Results - We collected data on the time it took participants to complete each section of the path, if and where they went wrong, and had them fill out a short survey on their impression of the interaction. The time it took to wait and go up the elevator was deducted from all the trials.

Five participants in the instructions-only case got lost and never made it to the destination, whereas all of the MRHG participants made it to the goal, a large difference in success rate of 66% vs. 100%. The participants who went wrong could not remember the entire sequence of instructions, and at some point they got lost and took a wrong turn.

In terms of timing, the *Leading* condition was the slowest with an average of 206 seconds. Next was the *Instructions* condition with 164 seconds on average. The fastest was the MRHG condition with 156 seconds on average. The difference in timing between the *Instructions* (164s) and the MRHG (156s) condition was not statistically significant (p=0.254). However, this does not account for the fact that 5 *Instructions* participants failed to reach the destination: we stopped timing their runs once they gave up. Note that this does not contrast our assumption that humans eventually reach the destination since we did not allow participants to ask for additional instructions.

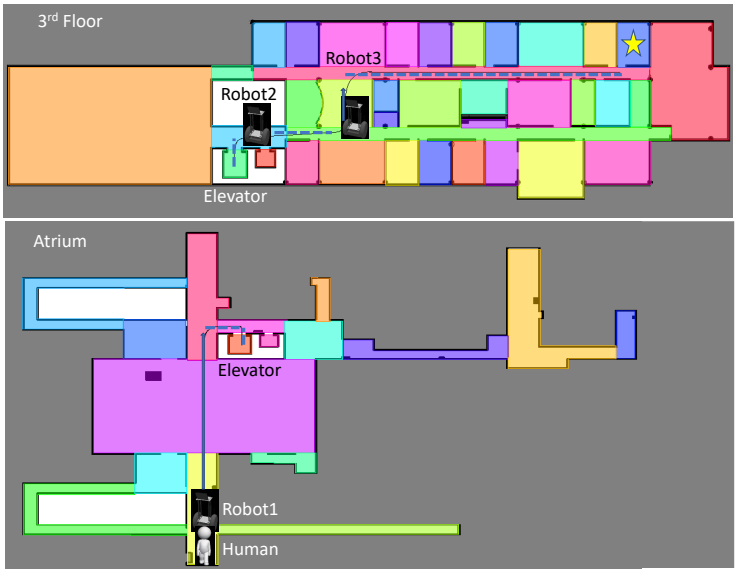

Fig. 2. In this example, a human approaches a robot in the atrium and asks for guidance to an office on the 3rd floor, marked with a star. The robot calculates a plan to optimally guide the human using two other robots through 10 regions. Solid lines represent *Lead* and dashed lines represent *Instruct* regions. Regions are colored uniquely.

To account for this effect, we add a penalty by assigning them the average time for the participants who got lost but kept looking and eventually reached the destination. For the adjusted *Instructions + Penalty* case we get an average of 183 seconds which is statistically significantly different from the MRHG condition *(p=0.026) for a two sample t-test. The *Leading* condition took 206 seconds on average and is statistically significantly difference from MRHG **(p=0.003).[2] A bar chart of the results is shown in Figure 3.

Participants responded to a post-interaction survey of their reactions to the interaction. The questions on the survey were chosen specifically to measure the utility of the robots in this task as well as users' enjoyment of the interaction. Questions were posed on a 5-point scale. The survey results were somewhat inconclusive. On the one hand, the MRHG participants ranked the robots as more friendly (p=0.064), useful (p=0.329), helpful **(p=0.003) and intelligent (p=0.085). They also ranked the interaction to feel more natural (p=0.06) than the *Instructions* participants ranked their interaction. On the other hand, they ranked the interaction to feel longer (p=0.054), and the instructions harder to remember **(p=0.000), follow **(p=0.000), and understand **(p=0.004) compared to the ranking provided by the *Instructions* participants. This result is surprising considering that 100% of the *Instructions* participants requested that the robot repeat the instructions and a third of them didn't make it

[2] We mark by * the paired measures for which there is significance at $p \leq 0.05$ and ** for measures at $p \leq 0.01$.

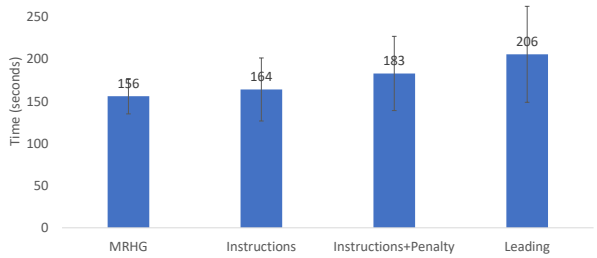

Fig. 3. Comparison of the average time for each guidance method. Error bars represent one standard deviation above and below the average.

to the destination. We emphasize that no participant witnessed both the cases, and conclude that more testing should be conducted to establish the perceived ease-of-use of the system. The survey results can be seen in Figure 4.

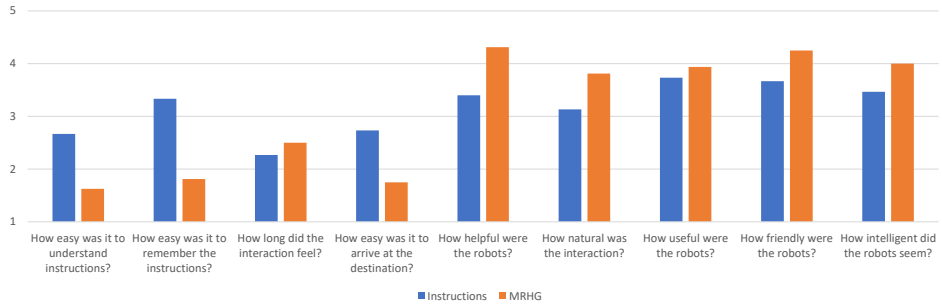

Fig. 4. Survey results of *MRHG* and *Instructions* participants.

8 Conclusions

In this study we developed a system for guiding humans in indoor environments that integrates multi-robot coordination with natural-language instruction generation, which, to the best of our knowledge, is the first system to do so. The system optimizes the guidance process by choosing the robots' locations, the regions through which they lead and the regions through which they instruct, in order to minimize the human's travel time. The instruction-generation module uses the robots' navigational path planner coupled with a landmark annotated map to generate natural language instructions.

The full MRHG system was tested on human participants and performed better than the *Instructions* benchmark in terms of both success rate and time to destination. It also outperformed the *Leading* benchmark in terms of time to destination.

This project showed, for the first time, that using natural language generation is beneficial in multi-robot human guidance systems. In this study we focused on

generating natural language instructions based on the robots' planned path which is the shortest path in terms of distance, but might not be the shortest path to guide a human through as there might be a longer path that is easier to guide through. Future work might consider extending this study by optimizing over all possible paths.

ACKNOWLEDGMENT

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-1637736, CPS-1739964, IIS-1724157), ONR (N00014-18-2243), FLI (RFP2-000), ARL, DARPA, and Lockheed Martin. Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References

1. D. Bohus, C. W. Saw, and E. Horvitz. Directions robot: in-the-wild experiences and lessons learned. In *AAMAS*, pages 637–644, 2014.
2. A. F. Daniele, M. Bansal, and M. R. Walter. Navigational instruction generation as inverse reinforcement learning with neural machine translation. In *HRI*, pages 109–118, 2017.
3. A. Denis. The loria instruction generation system l in give 2.5. In *The European Workshop on Natural Language Generation*, pages 302–306. Association for Computational Linguistics, 2011.
4. E. Eaton, C. Mucchiani, M. Mohan, D. Isele, J. M. Luna, and C. Clingerman. Design of a low-cost platform for autonomous mobile service robots. In *The IJCAI Workshop on Autonomous Mobile Service Robots*, 2016.
5. H. Hile, R. Grzeszczuk, A. Liu, R. Vedantham, J. Košecka, and G. Borriello. Landmark-based pedestrian navigation with enhanced spatial reasoning. In *International Conference on Pervasive Computing*, pages 59–76. Springer, 2009.
6. P. Khandelwal, S. Barrett, and P. Stone. Leading the way: An efficient multi-robot guidance system. In *AAMAS*, pages 1625–1633, 2015.
7. P. Khandelwal and P. Stone. Multi-robot human guidance: Human experiments and multiple concurrent requests. In *AAMAS*, pages 1369–1377, 2017.
8. P. Khandelwal, S. Zhang, J. Sinapov, M. Leonetti, J. Thomason, F. Yang, I. Gori, M. Svetlik, P. Khante, V. Lifschitz, et al. Bwibots: A platform for bridging the gap between ai and human–robot interaction research. *The International Journal of Robotics Research*, 36(5-7):635–659, 2017.
9. M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.
10. K. Striegnitz, A. Denis, A. Gargett, K. Garoufi, A. Koller, and M. Theune. Report on the second second challenge on generating instructions in virtual environments (give-2.5). In *The European workshop on natural language generation*, pages 270–279, 2011.
11. A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *SSW*, 125, 2016.
12. M. M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal. Cobots: Robust symbiotic autonomous mobile service robots. In *IJCAI*, page 4423. Citeseer, 2015.